A Matlab Spectral Integration Suite (SISMatlab)

Gokul Hariharan,[1,*] Satish Kumar,[2,†] and Mihailo R. Jovanović[1,‡]

[1] *Ming Hsieh Department of Electrical and Computer Engineering, University of Southern California, Los Angeles, CA 90089, USA*
[2] *Department of Chemical Engineering and Materials Science, University of Minnesota, Minneapolis, MN 55455, USA*

We provide a description of the spectral integration method and A Matlab Spectral Integration Suite (SISMatlab). Additional information about the examples provided in the paper along with Matlab source codes can be found at:

https://viterbi-web.usc.edu/∼mihailo/software/sismatlab/

* https://gokulhari.github.io/webpage/; gokulhar@usc.edu
† https://kumar.cems.umn.edu; kumar030@umn.edu
‡ https://viterbi-web.usc.edu/∼mihailo/; mihailo@usc.edu

I. THE SPECTRAL INTEGRATION METHOD: INTRODUCTION VIA AN EXAMPLE

We develop a customized Spectral Integration Suite in Matlab (SISMatlab) that is based on the method described in [2, 4]. In order to facilitate application to differential equations of order n, our implementation introduces minor modifications that we summarize next.

I.1. Basic features of SISMatlab

Let us consider a second-order linear differential equation,

$$\left(\mathrm{D}^2 \;+\; \frac{1}{y^2+1}\,\mathrm{D} \;-\; \epsilon^2 I\right)\phi(y) \;=\; d(y), \tag{1a}$$

with Dirichlet, Neumann, or Robin boundary conditions,

$$\phi(\pm 1) \;=\; \pm 1, \tag{1b}$$
$$[\mathrm{D}\phi(\cdot)](\pm 1) \;=\; \pm 2, \tag{1c}$$
$$4\,[\mathrm{D}\phi(\cdot)](\pm 1) \;+\; 3\,\phi(\pm 1) \;=\; \pm 3. \tag{1d}$$

Here, ϕ is the field of interest, d is an input, $\epsilon \in \mathbb{R}$ is a given constant, $y \in [-1, 1]$, and $\mathrm{D} := \mathrm{d}/\mathrm{d}y$.

In the spectral integration method, the highest derivative in a differential equation is expressed in a basis of Chebyshev polynomials. Specifically, for Eq. (1a), we have

$$\mathrm{D}^2\phi(y) \;=\; {\sum_{i=0}^{\infty}}' \phi_i^{(2)} T_i(y) \;=:\; \mathbf{t}_y^T \mathbf{\Phi}^{(2)}, \tag{2}$$

where $\sum'$ denotes a summation with the first term halved, $\mathbf{\Phi}^{(2)} := [\,\phi_0^{(2)} \;\; \phi_1^{(2)} \;\; \phi_2^{(2)} \;\; \cdots\,]^T$ is the infinite vector of spectral coefficients $\phi_i^{(2)}$, and $\mathbf{t}_y$ is the vector of Chebyshev polynomials of the first kind $T_i(y)$,

$$\mathbf{t}_y^T \;:=\; \left[\,\tfrac{1}{2}T_0(y) \;\; T_1(y) \;\; T_2(y) \;\; \cdots\,\right]. \tag{3}$$

Integration of Eq. (2) in conjunction with the recurrence relations for integration of Chebyshev polynomials are used to determine spectral coefficients corresponding to lower derivatives of ϕ. For example, indefinite integration of Eq. (2) yields

$$\mathrm{D}\,\phi(y) \;=\; {\sum_{i=0}^{\infty}}' \phi_i^{(1)} T_i(y) \;+\; c_1 \;=:\; \mathbf{t}_y^T \mathbf{\Phi}^{(1)} \;+\; c_1, \tag{4}$$

where c_1 is an integration constant and

$$\phi_i^{(1)} \;=\; \begin{cases} \frac{1}{2}\,\phi_1^{(2)}, & i \;=\; 0, \\ \frac{1}{2i}(\phi_{i-1}^{(2)} \;-\; \phi_{i+1}^{(2)}), & i \;\geq\; 1. \end{cases} \tag{5}$$

These expressions for $\phi_i^{(1)}$ are derived in Appendix A.

Similarly, indefinite integration of $\mathrm{D}\phi$ allows us to express ϕ as

$$\phi(y) \;=\; {\sum_{i=0}^{\infty}}' \phi_i^{(0)} T_i(y) \;+\; \tilde{c}_0 \;+\; c_1 y \;=:\; \mathbf{t}_y^T \mathbf{\Phi}^{(0)} \;+\; \tilde{c}_0 \;+\; c_1 y,$$

where $\tilde{c}_0$ and c_1 are integration constants and

$$\phi_i^{(0)} \;=\; \begin{cases} \frac{1}{2}\,\phi_1^{(1)}, & i \;=\; 0, \\ \frac{1}{2i}(\phi_{i-1}^{(1)} \;-\; \phi_{i+1}^{(1)}), & i \;\geq\; 1. \end{cases}$$

Equation (5) provides a recursive relation that is used to determine spectral coefficients of lower derivatives

from the spectral coefficients of the highest derivative of the variable ϕ,

$$\mathbf{\Phi}^{(1)} \;=\; \mathbf{Q}\,\mathbf{\Phi}^{(2)},\;\; \mathbf{\Phi}^{(0)} \;=\; \mathbf{Q}^2\,\mathbf{\Phi}^{(2)},$$

where $\mathbf{Q}^2 := \mathbf{Q}\,\mathbf{Q}$, and

$$\mathbf{Q} \;:=\; \begin{bmatrix} 0 & \frac{1}{2} & 0 & \cdots & & \\ \frac{1}{2} & 0 & -\frac{1}{2} & 0 & \cdots & \\ 0 & \frac{1}{4} & 0 & -\frac{1}{4} & 0 & \cdots \\ 0 & 0 & \frac{1}{6} & 0 & -\frac{1}{6} & 0 & \cdots \\ \vdots & \vdots & & \ddots & \ddots & \ddots & \end{bmatrix}. \tag{6}$$

Since $T_0(y) = 1$ and $T_1(y) = y$, we let $c_0 := 2\,\tilde{c}_0$ and represent integration constants in the basis expansion of ϕ and $\mathrm{D}\,\phi$ in terms of Chebyshev polynomials,

$$\phi(y) \;=\; \mathbf{t}_y^T\mathbf{Q}^2\,\mathbf{\Phi}^{(2)} \;+\; \begin{bmatrix} \frac{1}{2}T_0(y) & T_1(y) \end{bmatrix} \overbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}^{\mathbf{K}^0} \begin{bmatrix} c_0 \\ c_1 \end{bmatrix}, \tag{7}$$

$$\mathrm{D}\,\phi(y) \;=\; \mathbf{t}_y^T\mathbf{Q}\,\mathbf{\Phi}^{(2)} \;+\; \begin{bmatrix} \frac{1}{2}T_0(y) & T_1(y) \end{bmatrix} \underbrace{\begin{bmatrix} 0 & 2 \\ 0 & 0 \end{bmatrix}}_{\mathbf{K}^1} \begin{bmatrix} c_0 \\ c_1 \end{bmatrix}. \tag{8}$$

By introducing the vector of integration constants $\mathbf{c}^{(2)} := \begin{bmatrix} c_0 & c_1 \end{bmatrix}^T$, we can represent ϕ, $\mathrm{D}\phi$, and $\mathrm{D}^2\phi$ as

$$\phi(y) \;=\; \mathbf{t}_y^T\left(\mathbf{Q}^2\,\mathbf{\Phi}^{(2)} \;+\; \mathbf{R}_2\,\mathbf{c}^{(2)}\right) \;=\; \mathbf{t}_y^T \underbrace{\begin{bmatrix} \mathbf{Q}^2 & \mathbf{R}_2 \end{bmatrix}}_{\mathbf{J}_2} \begin{bmatrix} \mathbf{\Phi}^{(2)} \\ \mathbf{c}^{(2)} \end{bmatrix}, \tag{9}$$

$$\mathrm{D}\phi(y) \;=\; \mathbf{t}_y^T\left(\mathbf{Q}^1\,\mathbf{\Phi}^{(2)} \;+\; \mathbf{R}_1\,\mathbf{c}^{(2)}\right) \;=\; \mathbf{t}_y^T \underbrace{\begin{bmatrix} \mathbf{Q}^1 & \mathbf{R}_1 \end{bmatrix}}_{\mathbf{J}_1} \begin{bmatrix} \mathbf{\Phi}^{(2)} \\ \mathbf{c}^{(2)} \end{bmatrix}, \tag{10}$$

$$\mathrm{D}^2\phi(y) \;=\; \mathbf{t}_y^T\left(\mathbf{Q}^0\,\mathbf{\Phi}^{(2)} \;+\; \mathbf{R}_0\,\mathbf{c}^{(2)}\right) \;=\; \mathbf{t}_y^T \underbrace{\begin{bmatrix} \mathbf{Q}^0 & \mathbf{R}_0 \end{bmatrix}}_{\mathbf{J}_0} \begin{bmatrix} \mathbf{\Phi}^{(2)} \\ \mathbf{c}^{(2)} \end{bmatrix}, \tag{11}$$

where $\mathbf{Q}^0 = \mathbf{I}$ is an identity operator,

$$\mathbf{R}_i \;=\; \begin{bmatrix} \mathbf{K}^{2-i} \\ \mathbf{0} \end{bmatrix}, \qquad i \;=\; 0, 1, 2,$$

are matrices with an infinite number of rows and two columns, and $\mathbf{K}^2 = \mathbf{K}\,\mathbf{K}$ is a 2×2 zero matrix.

Finally, we utilize the expression for the product of two Chebyshev series [2, 6] to account for the non-constant coefficient $a(y) := 1/(y^2+1)$ in Eq. (1a). For a function $a(y)$ whose representation in the basis of Chebyshev polynomials is given by $a(y) = \sum_i{}' a_i\,T_i(y)$, the multiplication operator is given by

$$\mathbf{M}_a \;=\; \frac{1}{2}\begin{bmatrix} a_0 & a_1 & a_2 & a_3 & \cdots \\ a_1 & a_0 & a_1 & a_2 & \ddots \\ a_2 & a_1 & a_0 & a_1 & \ddots \\ a_3 & a_2 & a_1 & a_0 & \ddots \\ \vdots & \ddots & \ddots & \ddots & \ddots \end{bmatrix} \;+\; \frac{1}{2}\begin{bmatrix} 0 & 0 & 0 & 0 & \cdots \\ a_1 & a_2 & a_3 & a_4 & \cdots \\ a_2 & a_3 & a_4 & a_5 & \ddots \\ a_3 & a_4 & a_5 & a_6 & \ddots \\ \vdots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}. \tag{12}$$

Thus, in the basis of Chebyshev polynomials, we can express the differential equation (1a) as,

$$\mathbf{t}_y^T \left(\mathbf{J}_0 + \mathbf{M}_a \mathbf{J}_1 - \epsilon^2 \mathbf{J}_2\right) \begin{bmatrix} \mathbf{\Phi}^{(2)} \\ \mathbf{c}^{(2)} \end{bmatrix} = \mathbf{t}_y^T \mathbf{d}, \tag{13}$$

where $\mathbf{d}$ is the vector of spectral coefficients associated with the input $d(y)$ in Eq. (1a). Furthermore, we can use Eq. (9) to write the Dirichlet boundary conditions in Eq. (1b) as

$$\mathbf{t}_{\pm 1}^T \mathbf{J}_2 \begin{bmatrix} \mathbf{\Phi}^{(2)} \\ \mathbf{c}^{(2)} \end{bmatrix} = \pm 1, \tag{14a}$$

Eq. (10) to express the Neumann boundary conditions in Eq. (1c) as

$$\mathbf{t}_{\pm 1}^T \mathbf{J}_1 \begin{bmatrix} \mathbf{\Phi}^{(2)} \\ \mathbf{c}^{(2)} \end{bmatrix} = \pm 2, \tag{14b}$$

and Eqs. (9) and (10) to represent the Robin boundary conditions in Eq. (1d) as

$$\mathbf{t}_{\pm 1}^T \left(4\,\mathbf{J}_1 + 3\,\mathbf{J}_2\right) \begin{bmatrix} \mathbf{\Phi}^{(2)} \\ \mathbf{c}^{(2)} \end{bmatrix} = \pm 3. \tag{14c}$$

Combining Eqs. (13) and (14a), we can represent Eq. (1a) with Dirichlet boundary conditions (1b) in the basis of Chebyshev polynomials as

$$\underbrace{\begin{bmatrix} \mathbf{J}_0 + \mathbf{M}_a \mathbf{J}_1 - \epsilon^2 \mathbf{J}_2 \\ \mathbf{t}_{+1}^T \mathbf{J}_2 \\ \mathbf{t}_{-1}^T \mathbf{J}_2 \end{bmatrix}}_{\mathbf{F}} \underbrace{\begin{bmatrix} \mathbf{\Phi}^{(2)} \\ \mathbf{c}^{(2)} \end{bmatrix}}_{\mathbf{v}} = \underbrace{\begin{bmatrix} \mathbf{d} \\ +1 \\ -1 \end{bmatrix}}_{\mathbf{f}} \quad \Rightarrow \quad \mathbf{F}\,\mathbf{v} = \mathbf{f}. \tag{15}$$

Similarly, the problem with Neumann and Robin boundary conditions can be solved by replacing the last two rows in Eq. (15) with Eqs. (14b) and (14c), respectively.

We use the projection operator (see [6, Section 2.4])

$$\mathbf{P} = [\mathbf{I}_{N+1} \;\; \mathbf{0}], \tag{16}$$

where $\mathbf{I}_{N+1}$ is an identity matrix of dimension $N+1$, and $\mathbf{P}$ is a matrix with an infinite number of columns, to obtain a finite-dimensional approximation of Eq. (15) by truncating the infinite vector of spectral coefficients $\mathbf{\Phi}^{(2)}$ to a vector $\hat{\mathbf{\Phi}}^{(2)}$ with $N+1$ components,

$$\hat{\mathbf{F}}\,\hat{\mathbf{v}} = \hat{\mathbf{f}}, \tag{17}$$

where

$$\hat{\mathbf{F}} = \mathbf{S}\,\mathbf{F}\,\mathbf{S}^T, \quad \hat{\mathbf{v}} = \mathbf{S}\,\mathbf{v}, \quad \hat{\mathbf{f}} = \mathbf{S}\,\mathbf{f}, \quad \mathbf{S} = \begin{bmatrix} \mathbf{P} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_2 \end{bmatrix}. \tag{18}$$

Based on Eq. (9), an approximate solution $\hat{\phi}(y)$ to Eq. (1a) is obtained by integrating the solution $\hat{\mathbf{v}}$ to Eq. (17) twice,

$$\hat{\phi}(y) = \hat{\mathbf{t}}_y^T \hat{\mathbf{J}}_2 \hat{\mathbf{F}}^{-1} \hat{\mathbf{f}}, \tag{19}$$

where $\hat{\mathbf{t}}_y = \mathbf{P}\,\mathbf{t}_y$ and $\hat{\mathbf{J}}_2 = \mathbf{P}\,\mathbf{J}_2\,\mathbf{S}^T$.

I.2. Eigenvalue and frequency response analysis

Herein, we illustrate how the spectral integration method can be used to conduct modal and nonmodal analysis of the reaction-diffusion equation,

$$\phi_t(y,t) \;=\; \phi_{yy}(y,t) \;-\; \epsilon^2\phi(y,t) \;+\; d(y,t), \tag{20a}$$

with homogeneous Neumann boundary conditions,

$$[\partial_y\phi(\cdot,t)](\pm 1) \;=\; 0, \tag{20b}$$

where t is time, $y \in [-1,1]$ is a spatial variable, and $\epsilon \in \mathbb{R}$.

Eigenvalue decomposition. Let us consider the eigenvalue decomposition of the operator in system (20),

$$(\mathrm{D}^2 \;-\; \epsilon^2 I)\,\phi(y) \;=\; \lambda\,\phi(y), \tag{21a}$$

$$\mathrm{D}\,\phi(\pm 1) \;=\; 0, \tag{21b}$$

where λ is an eigenvalue and ϕ is an eigenfunction. Using the relations in Eqs. (9)-(11), differential equation (21a) and boundary conditions (21b) can be expressed in the basis of Chebyshev polynomials as,

$$\underbrace{\left(\mathbf{J}_0 \;-\; \epsilon^2\mathbf{J}_2\right)}_{\mathbf{F}}\underbrace{\begin{bmatrix}\boldsymbol{\Phi}^{(2)}\\ \mathbf{c}^{(2)}\end{bmatrix}}_{\mathbf{v}} \;=\; \lambda\,\mathbf{J}_2\underbrace{\begin{bmatrix}\boldsymbol{\Phi}^{(2)}\\ \mathbf{c}^{(2)}\end{bmatrix}}_{\mathbf{v}}, \tag{22a}$$

$$\underbrace{\begin{bmatrix}\mathbf{t}_{-1}\mathbf{J}_1\\ \mathbf{t}_{+1}\mathbf{J}_1\end{bmatrix}}_{\mathbf{M}}\underbrace{\begin{bmatrix}\boldsymbol{\Phi}^{(2)}\\ \mathbf{c}^{(2)}\end{bmatrix}}_{\mathbf{v}} \;=\; 0, \tag{22b}$$

or, equivalently,

$$\mathbf{F}\,\mathbf{v} \;=\; \lambda\,\mathbf{J}_2\,\mathbf{v}, \tag{23a}$$

$$\mathbf{M}\,\mathbf{v} \;=\; 0, \tag{23b}$$

Thus, only functions $\mathbf{v}$ that belong to the null-space of the operator $\mathbf{M}$ in Eq. (23b) are permissible solutions to Eq. (23a). A finite-dimensional representation of system (23) is given by

$$\hat{\mathbf{F}}\,\hat{\mathbf{v}} \;=\; \lambda\,\hat{\mathbf{J}}_2\,\hat{\mathbf{v}}, \tag{24a}$$

$$\hat{\mathbf{M}}\,\hat{\mathbf{v}} \;=\; 0, \tag{24b}$$

where,

$$\hat{\mathbf{F}} \;=\; \mathbf{P}\,\mathbf{F}\,\mathbf{S}^T, \quad \hat{\mathbf{M}} \;=\; \mathbf{M}\,\mathbf{S}^T, \quad \hat{\mathbf{J}}_2 \;=\; \mathbf{P}\,\mathbf{J}_2\,\mathbf{S}^T, \quad \hat{\mathbf{v}} \;=\; \mathbf{S}\,\mathbf{v},$$

with $\mathbf{P}$ and $\mathbf{S}$ defined in Eqs. (16) and (18), respectively. SVD of the fat full-row-rank matrix $\hat{\mathbf{M}}$ in (24b) can be used to parameterize its null-space [5],

$$\hat{\mathbf{M}}\,\hat{\mathbf{v}} \;=\; \hat{\mathbf{U}}\hat{\boldsymbol{\Sigma}}\hat{\mathbf{V}}^\dagger\hat{\mathbf{v}} \;=\; \hat{\mathbf{U}}\begin{bmatrix}\hat{\boldsymbol{\Sigma}}_1 & \mathbf{0}\end{bmatrix}\begin{bmatrix}\hat{\mathbf{V}}_1^\dagger\\ \hat{\mathbf{V}}_2^\dagger\end{bmatrix}\hat{\mathbf{v}} \;=\; 0. \tag{25}$$

Hence, $\hat{\mathbf{v}} := \hat{\mathbf{V}}_2\,\hat{\mathbf{u}}$ parametrizes the null-space [8] of the matrix $\hat{\mathbf{M}}$ in Eq. (24b). Substituting this expression for $\hat{\mathbf{v}}$ to Eq. (24a) yields the finite-dimensional generalized eigenvalue problem,

$$(\hat{\mathbf{F}}\hat{\mathbf{V}}_2)\,\hat{\mathbf{u}} \;=\; \lambda\,(\hat{\mathbf{J}}_2\hat{\mathbf{V}}_2)\,\hat{\mathbf{u}}, \tag{26}$$

which can be used to compute the eigenpair $(\lambda, \hat{\mathbf{u}})$. The eigenfunctions in Eq. (21) can be recovered by using

the null-space parametrization in conjunction with the integration operator (see Eq. (9)),

$$\hat{\phi}(y) \;=\; \hat{\mathbf{t}}_y^T\,\hat{\mathbf{J}}_2\,\hat{\mathbf{V}}_2\,\hat{\mathbf{u}}.$$

Frequency response analysis. The temporal Fourier transform can be used to represent the frequency response operator associated with system (20) as a two-point boundary value problem (TPBVP),

$$\begin{aligned}[\mathcal{A}(\omega)\,\phi(\cdot)]\,(y) &= [\mathcal{B}(\omega)\,d(\cdot)]\,(y),\\ \xi(y) &= [\mathcal{C}(\omega)\,\phi(\cdot)]\,(y),\\ [\mathcal{L}_a\,\phi(\cdot)](a) &= [\mathcal{L}_b\,\phi(\cdot)](b) \;=\; 0,\end{aligned} \tag{27}$$

where $\mathcal{A}(\omega) = (\mathrm{i}\omega+\epsilon^2)I - \mathrm{D}^2$, $\mathcal{B} = \mathcal{C} = I$, and $\mathcal{L}_{\pm 1} = \mathrm{D}$. A feedback interconnection of the frequency response operator with its adjoint can be used to compute singular values (see our paper and [1])

$$\begin{bmatrix} 0 & \mathcal{B}\mathcal{B}^\dagger \\ \mathcal{C}^\dagger\mathcal{C} & 0 \end{bmatrix}\begin{bmatrix}\phi(y)\\ \psi(y)\end{bmatrix} \;=\; \gamma\begin{bmatrix}\mathcal{A} & 0\\ 0 & \mathcal{A}^\dagger\end{bmatrix}\begin{bmatrix}\phi(y)\\ \psi(y)\end{bmatrix}, \tag{28a}$$

$$\begin{bmatrix}[\mathcal{L}_{\pm1}\phi(\cdot)](\pm1)\\ [\mathcal{L}_{\pm1}\psi(\cdot)](\pm1)\end{bmatrix} \;=\; \begin{bmatrix}0\\0\end{bmatrix}, \tag{28b}$$

where $(\cdot)^\dagger$ denotes the adjoint of the operator $(\cdot)$ and ψ is the auxiliary variable associated with the adjoint operator. The resulting eigenvalues determine the singular values in pairs of opposite signs, i.e., $\gamma = \pm\sigma$. Using definition of the operators $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, as well as their adjoints, we can express Eq. (28a) as

$$\begin{bmatrix} 0 & I\\ I & 0\end{bmatrix}\begin{bmatrix}\phi(y)\\ \psi(y)\end{bmatrix} \;=\; \lambda\begin{bmatrix}(\mathrm{i}\omega+\epsilon^2)I \;-\; \mathrm{D}^2 & 0\\ 0 & (-\mathrm{i}\omega+\epsilon^2)I \;-\; \mathrm{D}^2\end{bmatrix}\begin{bmatrix}\phi(y)\\ \psi(y)\end{bmatrix}, \tag{28c}$$

and employ Eqs. (9)-(11) to obtained a discretized approximation of Eq. (28),

$$\underbrace{\begin{bmatrix}\mathbf{0} & \mathbf{J}_2\\ \mathbf{J}_2 & \mathbf{0}\end{bmatrix}}_{\mathbf{F}}\underbrace{\begin{bmatrix}\boldsymbol{\Phi}^{(2)}\\ \mathbf{c}^{(\phi)}\\ \boldsymbol{\Psi}^{(2)}\\ \mathbf{c}^{(\psi)}\end{bmatrix}}_{\mathbf{v}} \;=\; \gamma\underbrace{\begin{bmatrix}(\mathrm{i}\omega+\epsilon^2)\mathbf{J}_2-\mathbf{J}_0 & \mathbf{0}\\ \mathbf{0} & (-\mathrm{i}\omega+\epsilon^2)\mathbf{J}_2-\mathbf{J}_0\end{bmatrix}}_{\mathbf{E}}\underbrace{\begin{bmatrix}\boldsymbol{\Phi}^{(2)}\\ \mathbf{c}^{(\phi)}\\ \boldsymbol{\Psi}^{(2)}\\ \mathbf{c}^{(\psi)}\end{bmatrix}}_{\mathbf{v}}, \tag{29a}$$

$$\underbrace{\begin{bmatrix}\mathbf{t}_{+1}^T\mathbf{J}_1 & \mathbf{0}\\ \mathbf{t}_{-1}^T\mathbf{J}_1 & \mathbf{0}\\ \mathbf{0} & \mathbf{t}_{+1}^T\mathbf{J}_1\\ \mathbf{0} & \mathbf{t}_{-1}^T\mathbf{J}_1\end{bmatrix}}_{\mathbf{M}}\begin{bmatrix}\boldsymbol{\Phi}^{(2)}\\ \mathbf{c}^{(\phi)}\\ \boldsymbol{\Psi}^{(2)}\\ \mathbf{c}^{(\psi)}\end{bmatrix} \;=\; \mathbf{0}. \tag{29b}$$

Finally, the projection operator (16) yields a finite dimensional approximation of system (29),

$$\hat{\mathbf{F}}\,\hat{\mathbf{v}} \;=\; \gamma\,\hat{\mathbf{E}}\,\hat{\mathbf{v}}, \tag{30a}$$

$$\hat{\mathbf{M}}\,\hat{\mathbf{v}} \;=\; \mathbf{0}, \tag{30b}$$

with

$$\hat{\mathbf{F}} \;=\; \mathbf{P}_2\,\mathbf{F}\,\mathbf{S}_2^T,\quad \hat{\mathbf{M}} \;=\; \mathbf{M}\,\mathbf{S}_2^T,\quad \hat{\mathbf{E}} \;=\; \mathbf{P}_2\,\mathbf{E}\,\mathbf{S}_2^T,\quad \hat{\mathbf{v}} \;=\; \mathbf{S}_2\mathbf{v},\quad \mathbf{P}_2 \;=\; \begin{bmatrix}\mathbf{P} & \mathbf{0}\\ \mathbf{0} & \mathbf{P}\end{bmatrix},\quad \mathbf{S}_2 \;=\; \begin{bmatrix}\mathbf{S} & \mathbf{0}\\ \mathbf{0} & \mathbf{S}\end{bmatrix},$$

where the expressions for $\mathbf{P}$ and $\mathbf{S}$ are given in Eqs. (16) and (18), respectively. The same procedure as in the modal analysis can be used to parametrize the null-space of the matrix $\hat{\mathbf{M}}$ in Eq. (30b). Namely, SVD

of $\hat{\mathbf{M}}$ in Eq. (25) reduces Eq. (30) to the generalized eigenvalue problem,

$$(\hat{\mathbf{F}}\,\hat{\mathbf{V}}_2)\,\hat{\mathbf{u}} \;=\; \gamma\,(\hat{\mathbf{E}}\,\hat{\mathbf{V}}_2)\,\hat{\mathbf{u}}, \tag{31}$$

yielding,

$$\begin{bmatrix} \hat{\phi}(y) \\ \hat{\psi}(y) \end{bmatrix} \;=\; \begin{bmatrix} \hat{\mathbf{t}}_y^T\,\hat{\mathbf{J}}_2 & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{t}}_y^T\,\hat{\mathbf{J}}_2 \end{bmatrix} \hat{\mathbf{V}}_2\,\mathbf{u}. \tag{32}$$

II. LINEAR DIFFERENTIAL OPERATORS WITH NON-CONSTANT COEFFICIENTS

In Section I, we described how the spectral integration method can be used to solve forced TPBVPs and conduct modal and nonmodal analyses of PDEs with one spatial variable and second-order differential operators. Herein, we illustrate the method for nth order linear differential operators with non-constant coefficients.

Let us consider an nth order linear differential equation with non-constant coefficients,

$$\sum_{k\,=\,0}^{n} a^{(k)}(y)\,\mathrm{D}^k\phi(y) \;=\; d(y), \tag{33a}$$

$$\sum_{k\,=\,0}^{n-1} b^{(k,\mathbf{p})}\,\mathrm{D}^k\phi(\mathbf{p}) \;=\; \mathbf{q}, \tag{33b}$$

where d is an input, ϕ is the unknown variable, $a^{(k)}(y)$ are non-constant coefficients, $b^{(k,\mathbf{p})}$ are constant coefficients associated with boundary constraints at a vector of evaluation points, $\mathbf{p}$, with the corresponding values at the boundaries, $\mathbf{q}$.

Differential equation. The highest derivative of ϕ in Eq. (33a) is expressed in a basis of Chebyshev polynomials as

$$\mathrm{D}^n\phi(y) \;=\; {\sum_{i\,=\,0}^{\infty}}' \phi_i^{(n)} T_i(y) \;=:\; \mathbf{t}_y^T\boldsymbol{\Phi}^{(n)}, \tag{34}$$

where $\boldsymbol{\Phi}^{(n)} := [\,\phi_0^{(n)}\;\;\phi_1^{(n)}\;\;\phi_2^{(n)}\;\;\cdots\,]^T$, and the lower derivatives are expressed as,

$$\mathrm{D}^i\phi(y) \;=\; \mathbf{t}_y^T\Big(\mathbf{Q}^{n-i}\boldsymbol{\Phi}^{(n)} \,+\, \mathbf{R}_{n-i}\,\mathbf{c}^{(n)}\Big) \;=\; \underbrace{\begin{bmatrix}\mathbf{Q}^{n-i} & \mathbf{R}_{n-i}\end{bmatrix}}_{\mathbf{J}_{n-i}} \begin{bmatrix} \boldsymbol{\Phi}^{(n)} \\ \mathbf{c}^{(n)} \end{bmatrix}. \tag{35}$$

Here, $\mathbf{Q}$ is defined in Eq. (6), $\mathbf{c}^{(n)} = [\,c_0\;\;c_1\;\;\cdots\;\;c_{n-1}\,]^T$ is the vector of integration constants, and

$$\mathbf{R}_i \;:=\; \begin{bmatrix} \mathbf{K}^{n-i} \\ \mathbf{0} \end{bmatrix},$$

are matrices with n columns and infinite number of rows with [4, Eq. 10]

$$\mathbf{K} \;=\; \begin{bmatrix} 0 & 2 & 0 & 6 & 0 & 10 & \cdots \\ 0 & 0 & 4 & 0 & 8 & 0 & \ddots \\ 0 & 0 & 0 & 6 & 0 & 10 & \ddots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix} \;\in\; \mathbb{R}^{n\times n}. \tag{36}$$

Infinite-dimensional representation and finite-dimensional approximation. Equations (12) and (35) can be used to express system (33) in the basis of Chebyshev polynomials as

$$\mathbf{t}_y^T \sum_{k=0}^{n} \mathbf{M}_{a^{(k)}}\, \mathbf{J}_{n-k} \begin{bmatrix} \mathbf{\Phi}^{(n)} \\ \mathbf{c}^{(n)} \end{bmatrix} \;=\; \mathbf{t}_y^T\, \mathbf{d}, \tag{37a}$$

$$\mathbf{t}_{\mathbf{p}}^T \sum_{k=0}^{n-1} b^{(k,\mathbf{p})}\, \mathbf{J}_{n-k} \begin{bmatrix} \mathbf{\Phi}^{(n)} \\ \mathbf{c}^{(n)} \end{bmatrix} \;=\; \mathbf{q}. \tag{37b}$$

Thus, a representation of the nth order differential equation (33a) with boundary conditions (33b) is given by,

$$\underbrace{\begin{bmatrix} \displaystyle\sum_{k=0}^{n} \mathbf{M}_{a^{(k)}}\, \mathbf{J}_{n-k} \\ \mathbf{t}_{\mathbf{p}}^T \displaystyle\sum_{k=0}^{n-1} b^{(k,\mathbf{p})}\, \mathbf{J}_{n-k} \end{bmatrix}}_{\mathbf{F}} \underbrace{\begin{bmatrix} \mathbf{\Phi}^{(n)} \\ \mathbf{c}^{(n)} \end{bmatrix}}_{\mathbf{v}} \;=\; \underbrace{\begin{bmatrix} \mathbf{d} \\ \mathbf{q} \end{bmatrix}}_{\mathbf{f}}. \tag{38}$$

Application of the projection operator given by Eq. (16) yields a finite-dimensional approximation,

$$\hat{\mathbf{F}}\, \hat{\mathbf{v}} \;=\; \hat{\mathbf{f}}, \tag{39}$$

where,

$$\hat{\mathbf{F}} \;=\; \mathbf{S}\, \mathbf{F}\, \mathbf{S}^T, \quad \hat{\mathbf{v}} \;=\; \mathbf{S}\, \mathbf{v}, \quad \hat{\mathbf{f}} \;=\; \mathbf{S}\, \mathbf{f}, \quad \mathbf{S} \;=\; \begin{bmatrix} \mathbf{P} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_n \end{bmatrix}. \tag{40}$$

The approximate solution $\hat{\phi}(y)$ to system (33) is obtained by integrating the solution $\hat{\mathbf{v}}$ to Eq. (39) n times,

$$\hat{\phi}(y) \;=\; \hat{\mathbf{t}}_y^T\, \hat{\mathbf{J}}_n\, \hat{\mathbf{F}}^{-1}\, \hat{\mathbf{f}},$$

where $\hat{\mathbf{t}}_y = \mathbf{P}\mathbf{t}_y$, $\hat{\mathbf{J}}_n = \mathbf{P}\mathbf{J}_n\mathbf{S}^T$, with $\mathbf{S}$ and $\mathbf{P}$ defined in Eqs. (40) and (16), respectively.

III. DESCRIPTION OF MATLAB FILES

SISMatlab contains a set of functions to obtain finite-dimensional approximations of linear operators via the spectral integration method. We next describe the functions that make up this suite.

Note: Our implementation requires N (where $N+1$ is the number of basis functions) to be an odd number.

- sety(N): sets $N+1$ points in physical space via $y_i = \cos(\pi\,(i+0.5)/(N+1))$. A discrete cosine transform of these points yields a vector that represents spectral coefficients in the basis of Chebyshev polynomials [7, Eq. 12.4.16-17].

```
N = 63; % N + 1 basis functions
y = sety(N); % spatially-independent variable
f = 1 - y.^2; % function in physical space
plot(y,f); % visualize results
```

- phys2cheb(f): takes a vector f in physical space (we refer to this as phys-space in this text and in our codes) and converts it to a vector of spectral coefficients in the basis of Chebyshev polynomials (we refer to this as cheb-space in this text and in our codes) via [7, Eq. 12.4.16-17].

```
N = 63; % N + 1 basis functions
y = sety(N); % spatially-independent variable
f_phys = 1 - y.^2; % [0.5*T_0; T_1; T_2]^T [1; 0; -0.5] = 0.5 - 0.5 (2 y^2 -1) = 1 - y^2
f_cheb = phys2cheb(f);
disp(f_cheb);
```

- cheb2phys.m: takes a vector of spectral coefficients in the basis of Chebyshev polynomials and coverts them to points in physical space via [7, Eq. 12.4.16-17].

- Matgen(n,N): generates [Q,K,J] = Matgen(n,N), where Q contains matrices that correspond to $\mathbf{Q}$ in Eq. (6), K to $\mathbf{K}$ in Eq. (36), and J to $\mathbf{J}$ in (35).

```
N = 63; % number of basis functions
m = 2;  % order of differential equation

% Use Matgen to represent Q, K, and J
[Q,K,J] = Matgen(m,N); % inputs: the order of the differential equation and N

% Q
Q{1} % Q^0: identity matrix
Q{2} % Q^1: matrix Q
Q{3} % Q^2: matrix Q*Q
% K
K{1} % K^1 matrix K
K{2} % K^0
% J
J{1} % J_0 matrix J
J{2} % J_1
J{3} % J_2
```

- MultMat(f): yields a finite-dimensional representation of the operator that accounts for spatially varying coefficients, i.e., the operator $\mathbf{M}_a$ in Eq. (12).

```
N = 63; % N + 1 basis functions
y = sety(N); % spatially-independent variable
% multiplication operator in the basis of Chebyshev polynomials
Ma = MultMat(1./(y.^2 + 1));
```

- Discretize(m,N,L): employs functions Matgen and MultMat to obtain finite-dimensional representations of linear operators. Inputs are the highest differential order, m, the number of basis functions, N, and the linear operator L. Linear operators are specified as cells in Matlab, e.g., the operator $a\,\mathrm{D}^2 + b\,\mathrm{D} + c$ is represented using a 3×1 cell with $L\{1\} = a$, $L\{2\} = b$, and $L\{3\} = c$).

  basic-example.m in the "examples" directory illustrates how to use Matgen and MultMat directly.

  Block-matrix operators are defined as cells of linear operators. For example,

```
% operator Dy
L11 = cell(2,1), L11{1} = 1; L11{2} = 0;

% operator Dyy + 2Dy
L12 = cell(3,1); L12{1} = 1; L12{2} = 2; L12{3} = 0;

% operator 2Dyy + 3Dy
L21 = cell(3,1); L21{1} = 2; L21{2} = 3; L21{3} = 1;

% Identity operator I
L22 = cell(1,1), L22{1} = 1;

% 2x2 block-matrix operator
L = cell(2,2);
L{1,1} = L11; L{1,2} = L12;
L{2,1} = L21; L{2,2} = L22;
```

- AdjointFormal(L): returns the formal adjoint of a linear operator or a block-matrix operator L.

```
Lad = AdjointFormal(L); % adjoint of the operator L
```

- MultOps(L1,L2): returns the composition of linear operators L_1 and L_2 of compatible dimensions.

```
L = MultOps(L1,L2); % composition of operators L1 and L2
```

- BcMat(n,N,eval,L): Generates a matrix of boundary evaluations given the highest order of the linear differential equation, n, N, the evaluation points, eval, and the linear operator to be applied at that point (Dirichlet, Neumann or mixed).

```
% Identity operator
I = cell(1,1); % cell array with coefficients
I{1} = 1.0;

% 1st derivative operator: 1.0 Dy + 0.0 I
Dy = cell(2,1); % cell array with coefficients
Dy{1} = 1.0; Dy{2} = 0.0;

% Operator 4 Dy + 3 I
Op = cell(2,1); % cell array with coefficients
Op{1} = 4; Op{2} = 3;

% boundary points
bcPts = [1; -1]; % \pm 1

% boundary operators
bcOp1 = {I; I}; % Dirichlet BCs
bcOp2 = {Dy; Dy}; % Neumann BCs
bcOp3 = {Op; Op}; % Robin BCs

% inputs to BcMat:
% differential order m, number of basis functions, N, boundary points, and boundary operator
bc1 = BcMat(m,N,bcPts,bcOp1); % Dirichlet BCs at \pm 1
bc2 = BcMat(m,N,bcPts,bcOp2); % Neumann BCs at \pm 1
bc3 = BcMat(m,N,bcPts,bcOp3); % Robin BCs at \pm 1
```

We also provide the following auxiliary functions that are useful in certain applications:

- ChebMat2CellMat(f,N): takes a matrix-valued function f of size $m(N+1) \times N$ and returns a cell of arrays of size $m \times N$. Each element in this cell is a vector that represents a function in y.

```
N = 63; % N + 1 basis functions
y = sety(N); % spatially-independent variable
% define entries of a matrix-valued function f
f11 = y; f12 = 1 - y.^2;
f21 = 1 + y.^3; f22 = y.^4;

% the matrix-valued function f
f = [f11, f12;
    f21, f22]; % matrix of dimension 128 x 2

f = ChebMat2CellMat(f,N); % convert f to a 2 x 2 cell
plot(y,f{1,2}); % visualize f12

% all functions in this format (cell of arrays) can be
% represented in the basis of Chebyshev polynomials
f_cheb = phys2cheb(f);
```

- integ(f): integrates a function f (defined in a physical space)

```
N = 63; % N + 1 basis functions
y = sety(N); % spatially-independent variable
% define a function f
f = 1 - y.^2;

% indefinite integral
f_indefinite = integ(f);

% integrate from -1 to 1
f_definite = val_rbc(f_indefinite) - val_lbc(f_indefinite);
```

  Functions val_rbc and val_lbc evaluate its argument in phys-space at $y = 1$ and $y = -1$ respectively.

- ChebEval(f,pts): evaluates a function defined in cheb-space at specified set of points in the domain.

```
N = 63; % N + 1 basis functions
y = sety(N); % spatially-independent variable
f = phys2cheb(1 - y.^2); % define a function f in the basis of Chebyshev polynomials

pts = [-1; 0.5; 1]; % set of points within a domain
values = ChebEval(f,pts); % evaluate function f at the specified set of points
```

We encourage you to try out the code snippets (compiled into a Matlab live-script, Illustrations_m in directory "examples") to see how the codes relate to this document. Furthermore, in directory "examples" we provide several scripts that illustrate the utility of SISMatlab; these may serve as templates for the use in your own problems. Examples whose file-names end with "_details" provide implementation details.

IV. USER-LEVEL FUNCTIONS

Functions described in Section III can be employed to solve a broad class of problems. For users who would like to use a simple interface, we also provide user-level functions for solving TPBVPs, conducting eigenvalue and singular value decompositions, and computing the H_∞ norm of a stable linear dynamical system.

- sisSolves(m,N,A,bc,d): solves a TPBVP of the form $[\mathcal{A}\phi(\cdot)](y) = d(y)$ with differential operator A of order m, boundary conditions bc, and input function d, where N determines the number of Chebyshev polynomials. By default, this function utilizes sparse matrix solvers; optional argument 'Full', i.e., sisSolves(m,N,A,bc,f,'Full'), specifies the use of dense linear algebra routines.

  The boundary conditions class BCs is used to specify boundary conditions bc. The number of boundary conditions and the number of variables are required inputs into BCs. For example, the following script can be used to solve

$$\begin{aligned}\left(\mathrm{D}^2 + \frac{1}{y^2+1}\,\mathrm{D} - \epsilon^2 I\right)\phi(y) &= d(y),\\ 2\,[\mathrm{D}\phi(\cdot)](-1) + 3\,\phi(-1) &= 4,\\ 5\,[\mathrm{D}\phi(\cdot)](+1) + 6\,\phi(+1) &= 7.\end{aligned}$$

```
% Set problem data
% number of basis functions, differential order, spatial variable, parameters
N = 63; % N has to be an odd number
m = 2; % order of differential operator
y = sety(N); % spatially-independent variable

% parameter eps in TPBVP
eps = 1; eps2 = eps*eps;

% Represent operator in TPBVP
A = cell(m+1,1); % cell array with coefficients
A{1} = 1.0; A{2} = 1./(y.^2 + 1); A{3} = -eps2;

% Input in physical space
d = 1 + y + y.^2;

% Robin BCs
bc = BCs(2,1); % two constraints on one variable

% boundary operators
OpLeft = cell(2,1); OpLeft{1} = 2; OpLeft{2} = 3; % for 2 Dy + 3 I
OpRight = cell(2,1); OpRight{1} = 5; OpRight{2} = 6; % for 5 Dy + 6 I

% Robin BCs at y = \pm 1
bc.Operator = {OpLeft; OpRight};
bc.Points = [-1; 1];
bc.Values = [4; 7];

% Use sisSolves to solve TPBVP
% inputs to sisSolves: differential order m, N, operator, BCs, input forcing d
solution = sisSolves(m,N,A,bc,d);
```

- sisEigs(m,N,F,E,bc): solves the generalized eigenvalue problem $[\mathcal{F}\phi(\cdot)](y) = \lambda\,[\mathcal{E}\phi(\cdot)](y)$, where $\mathcal{F}$ and $\mathcal{E}$ are differential operators in y. This function can be called via:

```
% solve generalized e-value problem via sisEigs
[V,lambda] = sisEigs(m,N,F,E,bc);
[V,lambda] = sisEigs(m,N,F,E,bc,K);
[V,lambda] = sisEigs(m,N,F,E,bc,K,SIGMA);
```

  Here, V and lambda are the eigenvectors and eigenvalues and m is a vector that specifies the differential order of each variable in the block-matrix operator, N is the number of Chebyshev polynomials, F and E are linear matrix-valued operators in the generalized eigenvalue problem, and bc are boundary conditions that are specified using class BCs.

  By default, this function computes 6 eigenvalues with the largest magnitude along with the corresponding eigenvectors. The optional arguments modify this behavior as done in Matlab's function eigs:

  - K – specifies number of eigenvalues to be computed;
  - SIGMA – if SIGMA is a scalar, the closest eigenvalues to SIGMA are computed; if SIGMA is a character array, use 'LR' and 'SR' for the eigenvalues with largest and smallest real part, and 'LM' and 'SM' for the eigenvalues with largest and smallest magnitude.

  Setting SIGMA to 'Full' signals the use of Matlab's function eig instead of eigs; in this special case, K is not used and all eigenpairs of the finite-dimensional approximation are computed.

- sisSvdfrs(m,mad,N,A,B,C,bcReg,bcAdj): computes singular value decomposition of the frequency response operator that can be represented via a TPBVP. This function can be called via:

```
% resolvent analysis via sisSvdfrs
% (i.e., SVD of the frequency response operator)
[V,gamma] = sisSvdfrs(m,mad,N,A,B,C,bcReg,bcAdj);
[V,gamma] = sisSvdfrs(m,mad,N,A,B,C,bcReg,bcAdj,K);
[V,gamma] = sisSvdfrs(m,mad,N,A,B,C,bcReg,bcAdj,K,SIGMA);
```

  Here, V contains the right and left singular functions of the feedback interconnected system (i.e., ϕ and ψ in (28)) and gamma is the vector of the singular values. The inputs to sisSvdfrs are: m and mad are vectors that specify the differential order of regular and adjoint variables in TPBVP, N is the number of Chebyshev polynomials, A, B, and C are matrix-valued operators $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ in the TPBVP representation given by Eq. (27), and bcReg and bcAdj are boundary conditions that regular and adjoint variables have to satisfy (these are to be specified using class BCs).

  By default, this function computes 6 singular values with the largest magnitude along with the corresponding right and left singular functions. The optional arguments modify this behavior as done in Matlab's function eigs:

  - K – specifies number of singular values to be computed;
  - SIGMA – if SIGMA is a scalar, the closest singular values to SIGMA are computed; if SIGMA is a character array, use 'LR' and 'SR' for the singular values with largest and smallest real part.

  Setting SIGMA to 'Full' signals the use of Matlab's function eig instead of eigs; in this special case, K is not used and all eigenpairs of the finite-dimensional approximation are computed.

  Note: We currently do not provide a routine to compute adjoint boundary conditions; you need to explicitly provide it as an input. However, we understand that this can be cumbersome as adjoint boundary conditions are derived by integrating by parts. We developed a Mathematica package 'AdjointFinder' that is posted on our website, and can help derive analytical expressions for adjoint boundary conditions (and operators) by automatic integration by parts.

- sisHinf(m,N,E,F,B,C,bcReg,bcAdj): Computes the H_∞ norm of a stable linear dynamical system,

$$\begin{aligned} \partial_t[\mathcal{E}\,\boldsymbol{\phi}(\cdot,t)](y) &= [\mathcal{F}\,\boldsymbol{\phi}(\cdot,t)](y) + [\mathcal{B}\,\boldsymbol{d}(\cdot,t)](y), && \text{(41a)}\\ \boldsymbol{\xi}(y,t) &= [\mathcal{C}\,\boldsymbol{\phi}(\cdot,t)](y), && \text{(41b)}\\ [\mathcal{L}_a\,\boldsymbol{\phi}(\cdot,t)](a) &= [\mathcal{L}_b\,\boldsymbol{\phi}(\cdot,t)](b) = 0. && \text{(41c)} \end{aligned}$$

The H_∞ norm is determined by the peak (over temporal frequency) of the largest singular value of the frequency response operator associated with system Eq. (41c), i.e., $\sup_\omega \mathcal{T}(\omega)$. This function can be called via:

```
% H infinity norm of a stable linear system via sisHinf
[wopt,Hinf] = sisHinf(m,N,E,F,B,C,bcReg,bcAdj);
```

where wopt is the frequency at which $\sup_\omega \mathcal{T}(\omega)$ takes place, Hinf is the H_∞ norm, m is a vector that specifies the differential order of each variable in the vector $\boldsymbol{\phi}$, N is the number of Chebyshev polynomials, E, F, B, and C are the operators in Eq. (41), and bcReg and bcAdj are boundary conditions that regular and adjoint variables have to satisfy (these are to be specified using class BCs).

Appendix A: Recurrence relations

Consider the expression for the highest derivative of a second order differential equation in a Chebyshev basis,

$$\mathrm{D}^2 u(y) = u_0^{(2)}\tfrac{1}{2}T_0(y) + u_1^{(2)}T_1(y) + u_2^{(2)}T_2(y) + u_3^{(2)}T_3(y) + \cdots. \tag{A1}$$

The Chebyshev polynomials are related to their derivatives via [3, Equation 3.25]

$$\begin{aligned} T_0(y) &= T_1'(y), && \text{(A2a)}\\ T_1(y) &= \tfrac{1}{4}T_2'(y), && \text{(A2b)}\\ T_n(y) &= \tfrac{1}{2}\left(\frac{T_{n+1}'(y)}{n+1} - \frac{T_{n-1}'(y)}{n-1}\right), \qquad n>1. && \text{(A2c)} \end{aligned}$$

Substitution of Eq. (A2) to Eq. (A1) along with an indefinite integration on the resulting expression yields,

$$\begin{aligned} \mathrm{D}\,v(y) = {}& \frac{u_0^{(2)}}{2}y + \frac{u_1^{(2)}}{2}y^2 + \frac{u_2^{(2)}}{2}\left(\frac{T_3(y)}{3} - \frac{T_1(y)}{1}\right) + \frac{u_3^{(2)}}{2}\left(\frac{T_4(y)}{4} - \frac{T_2(y)}{2}\right)\\ & + \frac{u_4^{(2)}}{2}\left(\frac{T_5(y)}{5} - \frac{T_3(y)}{3}\right) + \cdots + c_0, \end{aligned} \tag{A3}$$

where c_0 is the effective integration constant. Since $y^2 = (T_0(y)+T_2(y))/2$, Eq. (A3) takes the form:

$$\begin{aligned} \mathrm{D}\,v(y) = {}& \frac{u_0^{(2)}}{2}y + \frac{u_1^{(2)}}{2}\left(\frac{T_0(y)+T_1(y)}{2}\right) + \frac{u_2^{(2)}}{2}\left(\frac{T_3(y)}{3} - \frac{T_1(y)}{1}\right)\\ & + \frac{u_3^{(2)}}{2}\left(\frac{T_4(y)}{4} - \frac{T_2(y)}{2}\right) + \frac{u_4^{(2)}}{2}\left(\frac{T_5(y)}{5} - \frac{T_3(y)}{3}\right) + \cdots + c_0, \end{aligned} \tag{A4}$$

$$= T_1(y)\underbrace{\left(\frac{u_0^{(2)}}{2} - \frac{u_2^{(2)}}{2}\right)}_{u_1^{(1)}} + T_2(y)\underbrace{\left(\frac{u_1^{(2)}}{4} - \frac{u_3^{(2)}}{4}\right)}_{u_2^{(1)}} + T_3(y)\underbrace{\left(\frac{u_2^{(2)}}{6} - \frac{u_2^{(2)}}{6}\right)}_{u_3^{(1)}} + \cdots + \underbrace{\frac{u_1^{(2)}}{4}}_{u_0^{(1)}/2} + c_0. \tag{A5}$$

Hence, from Eq. (A5) we have,

$$\mathrm{D}\, v(y) \;=\; u_0^{(1)}\, \tfrac{1}{2} T_0(y) \;+\; u_1^{(1)}\, T_1(y) \;+\; u_2^{(1)}\, T_2(y) \;+\; u_3^{(1)}\, T_3(y) \;+\; \cdots \;+\; c_0, \tag{A6}$$

where $u_0^{(1)} = u_1^{(2)}/2$ and the remaining coefficients for $u_i^{(1)}$ can be obtained using the recursive relation in Eq. (5).

---

[1] S. Boyd, V. Balakrishnan, and P. Kabamba. A bisection method for computing the $\mathcal{H}_\infty$ norm of a transfer matrix and related problems. *Math. Control Signals Syst.*, 2(3):207–219, 1989.

[2] K. Du. On well-conditioned spectral collocation and spectral methods by the integral reformulation. *SIAM J. Sci. Comp.*, 38:A3247–A3263, 2016.

[3] A. Gil, J. Segura, and N. M. Temme. Chebyshev expansions. In *Numerical Methods for Special Functions*, chapter 3, pages 51–86. Society for Industrial and Applied Mathematics, 2007.

[4] L. Greengard. Spectral integration and two-point boundary value problems. *SIAM J. Numer. Anal.*, 28:1071–1080, 1991.

[5] M. R. Jovanović and B. Bamieh. A formula for frequency responses of distributed systems with one spatial variable. *Syst. Control Lett.*, 55(1):27–37, January 2006.

[6] S. Olver and A. Townsend. A fast and well-conditioned spectral method. *SIAM Rev.*, 55:462–489, 2013.

[7] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, 2007.

[8] G. Strang. The fundamental theorem of linear algebra. *Amer. Math. Monthly*, 100(9):848–855, 1993.